

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平6-214600

(43)公開日 平成6年(1994)8月5日

(51)Int.Cl.⁵

G 1 0 L 9/18
9/14

識別記号

E 8946-5H
J 8946-5H

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数22 F D (全 25 頁)

(21)出願番号 特願平5-342034

(22)出願日 平成5年(1993)12月14日

(31)優先権主張番号 9 9 0 3 0 9

(32)優先日 1992年12月14日

(33)優先権主張国 米国 (US)

(71)出願人 390035493

アメリカン テレフォン アンド テレグ
ラフ カムパニー

AMERICAN TELEPHONE
AND TELEGRAPH COMPA
NY

アメリカ合衆国 10013-2412 ニューヨ
ーク ニューヨーク アヴェニュー オブ
ジ アメリカズ 32

(72)発明者 ウイレム バスチアン クレイジュン

アメリカ合衆国、07920 ニュージャージ
ー、 バスキング リッジ、ピレッジ ド
ライブ 87

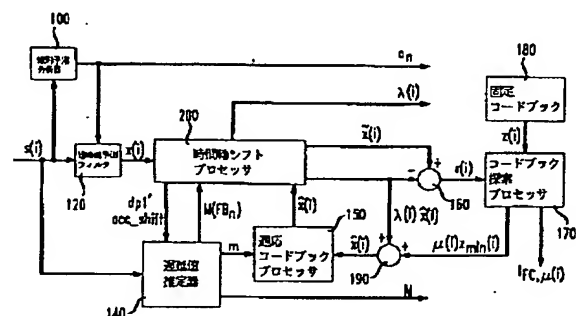
(74)代理人 弁理士 三俣 弘文

(54)【発明の名称】 汎用合成による分析符号化の時間軸シフト方法とその装置

(57)【要約】

【目的】 汎用合成による分析技術を用いた音声符号化方法を提供する。

【構成】 極大エネルギーを含む、原信号の一部が識別される。極大エネルギーを含む、原信号の複数のセグメントが複数の時間軸シフトに基づいて選択される。これらのセグメントは、「試原信号」と名付けられる。各試原信号は、適応コードブックから得られた合成信号と比較され、これらの信号間の類似度（例えば、相互相関）が評価される。符号化に使用される試原信号は、の評価類似度に基づいて決定される。原信号の符号化表現をもたらす信号が、決定試原信号に基づいて生成される。試原信号の符号化表現をもたらす信号は、符号励振形線形予測符号器のような合成による分析符号器により供給される。



【特許請求の範囲】

【請求項1】 a. 標本識別基準に基づいて、原信号の標本を識別するステップと、

b. 試原信号を形成する為に、識別された標本のうちの1個以上の標本を含む、原信号のセグメントを選択するステップと、

c. 複数の試原信号のそれぞれについて、試原信号と合成信号との類似度を評価するステップと、

d. 前記の評価された類似度に基づいて、符号化に使用される試原信号を決定するステップと、

e. 前記の決定された試原信号に基づいて、原信号の符号化表現を表す信号を生成するステップとからなる、汎用合成による分析符号化の時間軸シフト方法。

【請求項2】 1. 該試原信号を表現するパラメータを生成する為に、試原信号を分析するステップと、

2. 上記パラメータのうちの1個以上のパラメータに基づいて、原信号を推定する信号を合成するステップとを更に含むことを特徴とする請求項1記載の方法。

【請求項3】 前記(a)ステップは、原信号を分析することにより、極大エネルギーの位置を決定するステップを含むことを特徴とする請求項1記載の方法。

【請求項4】 原信号の選択されたセグメントは、識別された信号標本以外の原信号標本からなることを特徴とする請求項1記載の方法。

【請求項5】 選択されたセグメントは、他の原信号標本に先行する識別された標本からなることを特徴とする請求項4記載の方法。

【請求項6】 前記(b)ステップは、

1. 原信号の1個以上の標本に関して時間軸シフトを決定するステップと、

2. 上記時間軸シフトに基づいて、原信号標本集合を決定するステップとを含むことを特徴とする請求項1記載の方法。

【請求項7】 前記(c)ステップは、試原信号と合成信号との相互相関を形成するステップを含むことを特徴とする請求項1記載の方法。

【請求項8】 前記(d)ステップは、評価された複数の類似度の比較に基づいて、上記複数の試原信号の中から試原信号を選択するステップを含むことを特徴とする請求項1記載の方法。

【請求項9】 前記(d)ステップは、

(f) 評価された複数の類似度の比較に基づいて、試原信号を生成するステップを含むことを特徴とする請求項1記載の方法。

【請求項10】 前記(f)ステップは、

1. 複数の試原信号類似度の中から、実質的最大類似度を決定するステップと、

2. 実質的最大類似度をもたらす時間軸シフトを決定するステップとを含むことを特徴とする請求項9記載の方法。

【請求項11】 前記(f)ステップは、

形成された試原信号と時間軸シフトとに基づいて、試原信号用標本値を決定するステップを含むことを特徴とする請求項9記載の方法。

【請求項12】 前記(f)ステップは、

原信号と時間軸シフトとに基づいて、試原信号用標本値を決定するステップを含むことを特徴とする請求項10記載の方法。

【請求項13】 前記(e)ステップは、

(g) 1個以上の決定された試原信号を符号化するステップを含むことを特徴とする請求項1記載の方法。

【請求項14】 前記(g)ステップは、

(h) 合成による分析符号化を行うステップを含むことを特徴とする請求項13記載の方法。

【請求項15】 前記(h)ステップは、

符号励振形線形予測符号化を行うステップを含むことを特徴とする請求項14記載の方法。

【請求項16】 a. 標本識別基準に基づいて、原信号の標本を識別する手段と、

b. 試原信号を形成する為に、識別された標本のうちの1個以上の標本を含む、原信号のセグメントを選択する手段と、

c. 複数の試原信号のそれぞれについて、試原信号と合成信号との類似度を評価する手段と、

d. 前記の評価された類似度に基づいて、符号化に使用される試原信号を決定する手段と、

e. 前記の決定された試原信号に基づいて、原信号の符号化表現を表す信号を生成する手段とからなる、汎用合成による分析符号化の時間軸シフト装置。

【請求項17】 1. 該試原信号を表現するパラメータを生成する為に、試原信号を分析する手段と、

2. 上記パラメータのうちの1個以上のパラメータに基づいて、原信号を推定する信号を合成する手段とを更に含むことを特徴とする請求項1記載の装置。

【請求項18】 前記(a)手段は、

原信号を分析することにより、極大エネルギーの位置を決定する手段を含むことを特徴とする請求項16記載の装置。

【請求項19】 前記(b)手段は、

1. 原信号の1個以上の標本に関して時間軸シフトを決定する手段と、

2. 上記時間軸シフトに基づいて、原信号標本集合を決定する手段とを含むことを特徴とする請求項16記載の装置。

【請求項20】 前記(e)手段は、

(g) 1個以上の決定された試原信号を符号化する手段を含むことを特徴とする請求項16記載の装置。

【請求項21】 前記(g)手段は、

(h) 合成による分析符号化を行う手段を含むことを特徴とする請求項20記載の装置。

【請求項22】 前記(h)手段は、符号励振形線形予測符号化を行う手段を含むことを特徴とする請求項21記載の装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、一般的に音声符号化システムに関し、特に、合成による分析音声符号化システムにおける必要帯域幅の削減に関する。

【0002】

【従来の技術】音声符号化システムは、システム受信器に、チャンネルまたは通信網を通じての通信のための音声信号コードワード表現を提供するように機能する。各システム受信器は、受信したコードワードから音声信号を再構成する。与えられた期間内にシステムにより伝達されるコードワード情報量は、システム帯域幅の限界を定め、システム受信器が再生する音声品質に影響を及ぼす。

【0003】音声符号化システムの設計者は、しばしば、できるだけ狭い帯域幅を使用して、高品質の音声再生能力を提供しようとする。しかし、高品質音声を求めることと帯域幅を狭くすることとは、矛盾する虞があるので、設計過程においてエンジニアリングのトレードオフとなりうる。しかし、これにもかかわらず、狭いチャンネル帯域幅で合格音声品質を与える音声符号化技術がこれまで開発されている。この中に、合成による分析音声符号化技術がある。合成による分析音声符号化技術によれば、音声信号は、波形合成手法を通して符号化される。候補音声信号は、符号化しようとする原音声信号との比較のため、1個以上のパラメータから合成される。パラメータを変化させることにより、種々の合成された候補音声信号が決定される。ついで、最も類似した候補音声信号のパラメータが原音声信号を表現するのに使用される。

【0004】多数の合成による分析符号器（例えば、大部分の符号励振形線形予測(CELLP)符号器）は、音声信号中の長時間相関をモデル化する長時間予測器(LTP)を使用する。「音声信号」という語は、実音声または合成による分析符号器中に存在する残差信号および励振信号を意味する。合成過程中、長時間予測器は、通常、全極形フィルタまたは利得基準化付適応コードブックとして実現される。一般的事柄として、音声信号中の長時間相関は、過去に再構成された音声信号が現時音声信号の近似信号として機能するのを可能とする。長時間予測器は、(すでに符号化されている)個々の過去音声信号を現時(原)音声信号と比較するように動作する。このような比較により、長時間予測器は、いずれの過去音声信号が原音声信号に最近似するかを決定する。過去音声信号は、該過去音声信号が(現時刻から)どの程度の過去に見出されるかを示す遅延値によって識別される。長時間予測器を使用する符号器は、現時音声信号か

ら最も整合度の高い音声信号(すなわち、最良近似信号)の基準化された版を差引くことにより、長時間相関の小さい信号を生じる。ついで、この信号は、代表的な場合、固定確率コードブック(FSCB)を使用して、符号化される。固定確率コードブックの指標および長時間予測器遅延値は、他のパラメータと一緒に、これらのパラメータから推定原音声を復元しうる符号励振形線形予測復号器へ伝送される。

【0005】

【発明が解決しようとする課題】音声の長時間相関をモデル化することにより、復号器での再構成音声品質は、改善されうる。しかし、この改善は、帯域幅の有意の拡大なしには達成されない。例えば、音声の長時間相関をモデル化するために、通常符号励振形線形予測符号器は、5または7.5ms(サブフレームという)毎に8ビット遅延情報を伝送しうる。このような時間変化形遅延パラメータは、帯域幅1秒毎に、例えば、1~2kbを余分に必要とする。長時間予測器遅延値の変動は、時間全体に亘って予測されえない(すなわち、長時間予測器遅延値列は、本質的に確率的である)ので、改良された遅延パラメータ符号化により追加必要帯域幅を減らすのは、困難であることが判る。

【0006】長時間予測器を使用する合成による分析の追加必要帯域幅削減の一方策は、長時間予測器遅延値を伝送する頻度を減し、補間により中間長時間予測器遅延値を決定することであろう。しかし、補間は、音声信号の個別サブフレームにおいて、長時間予測器により最適と言えない遅延値が使用されることになる虞がある。例えば、遅延値が最適と言えないときは、長時間予測器は、最適と言えない方法で過去音声信号を現時点に写像することになる。結果として、現時点に写像された過去音声信号と原音声信号との差は、他の手段による場合のものに比べて大きくなる。この場合、固定確率コードブックは、波形を改良するという正常機能を行うよりも、上記の最適と言えない時間軸シフトの効果を無効とするように動作する。結果として、有意可聴歪みが生じうる。

【0007】本発明は、合成による分析の符号化システムにおいて、必要帯域幅を減らす方法および装置を提供する。

【0008】

【課題を解決するための手段】本発明によれば、汎用合成による分析符号化は、原信号を変化させることにより提供される。原信号の異形群を試原信号という。合成による分析符号化において、原信号の代りに、または、原信号使用の補充として、試原信号を使用するので、符号化誤差および必要ビットレートが削減される。音声符号化において、符号化誤差が小さくなれば、長時間予測器遅延情報の伝送頻度を低くすることができ、再構成音声品質の低下がほとんど無い、または、全く無い、遅延値

補間が可能となる。本発明は、とりわけ、無線電話網（例えば、セルラー式）および通常電話網のような音声情報通信網に適用されうる。

【0009】音声符号化に関して、試原信号は、説明として、知覚（例えば、聴覚）的に実原信号に類似した信号である。試原信号と実原信号との聴覚的類似度は、符号化ビットレートおよび受信器により合成される音声の品質に影響しうる（例えば、類似度が低いほど、ビットレートも低くなり、音声品質も低くなる虞がある）。原信号（したがって、試原信号）は、実原音声信号、または、合成による分析符号器中に存在する残差もしくは励振信号の形を取りうる。

【0010】本発明の実施例によれば、試原信号は、実音声信号セグメントの時間軸シフト版として生成される。試原信号と適応コードブックの補助情報との類似度（例えば、相互相関）が評価される。全試原信号のひとつ、または、原もしくは試原信号の異形と同一の試原信号が1個以上の評価類似度に基づいて決定される。先に生成された諸試原信号のうちのひとつの異形の場合には、決定された試原信号（すなわち、上記異形）は、先に生成された諸試原信号を生成した諸時間軸シフトに対応する。原信号の符号化表現をもたらす信号が決定された上記試原信号に基づいて生成される。

【0011】

【実施例】

実施例のハードウェア

説明を分かり易くするため、本発明の一実施例は、

（「プロセッサ」と名づけられた機能ブロックを含む）＊

$$x(i) = s(i) - \sum_{n=1}^N a_n s(i-n)$$

【0015】数式中、パラメータ a_n は、線形予測分析器10により供給される。Nは、（8kHzの標準化率について）通常、標準約10個であるから、励振信号 $x(i)$ は、一般的に、原信号 $s(i)$ の短時間周期性を保持する。この冗長性を除去するために、長時間予測器30が設けられている。

【0016】 $x(i)$ の値は、通常、ブロック基準で決定される。各ブロックは、サブフレームと名づけられている。線形予測係数 a_n は、線形予測分析器10によ

$$r(i) = x(i) - \lambda(i) \hat{x}(i-d(i)),$$

【0018】数式中、山付 $x(i-d(i))$ は、先のサブフレーム内で合成された（すなわち再構成された）音声信号の標本である。このようにして、長時間予測器30は、量 $\lambda(i) \cdot$ 山付 $x(i-d(i))$ を与える。信号 $r(i)$ は、 $\lambda(i) \cdot$ 山付 $x(i-d(i))$ が $x(i)$ から差引かれた後に残る励振信号である。ついで、信号 $r(i)$ は、固定確率コードブック40を使用して符号化される。固定確率コードブック40は、コードブックベクトル指標 $\mu(i)$ と同伴基準化

＊個々の機能ブロックからなるものとして示される。これらのブロックが表わす機能は、（ソフトウェアを実行しうるハードウェアを含むが、このハードウェアに限定されない）共用ハードウェアまたは専用ハードウェアの使用により実現されうる。例えば、図4および図6に示されたプロセッサの機能は、1個の共用プロセッサによって与えられうる。明細書中、「プロセッサ」という用語の使用は、ソフトウェアを実行する能力を有するハードウェアのみを指すものと解釈されるべきでない。

【0012】本発明の実施例は、AT&T DSP16もしくはDSP32Cのような（DSP）ハードウェア、後述する動作を遂行するソフトウェアを格納する読出し専用メモリ（ROM）、およびデジタル信号プロセッサの結果を記憶するランダムアクセスメモリ（RAM）からなる。超大規模集積回路（VLSI）ハードウェアの実施例が汎用デジタル信号プロセッサ回路と組合わせたカスタム超大規模集積回路と同様に提供される。

【0013】従来の符号励振形線形予測に関する説明

従来の合成による分析用符号励振形線形予測符号器が図14に示されている。標準化された音声信号 $s(i)$ （ i は、標本指標である）は、現時音声セグメントに対して最適化されたN次の短時間線形予測フィルタ（STP）20に与えられる。信号 $x(i)$ は、短時間線形予測フィルタによるろ波後に得られた励振である。

【0014】

【数1】

(1)

※り、フレーム毎基準で決定される。フレームは、一般的に、サブフレーム持続時間の整数倍である一定持続時間を有し、通常20～30msの長さである。 a_n のサブフレーム値は、通常、補間により決定される。（代表的な場合、適応コードブックにより実現される）長時間予測器は、使用利得 $\lambda(i)$ と使用遅延値 $d(i)$ とを以下のように決定する。

【0017】

【数2】

(2)

因数とをを生じる。これらの量は、一体となって、最近似の励振 $r(i)$ を与える。

【0019】各音声サブフレームを表わすデータ、すなわち、長時間予測器パラメータ $\lambda(i)$ および $d(i)$ と固定確率コードブック指標とは、フレームに等しくなる整数個のサブフレームについて集められる。上記データフレームは、線形予測係数 a_n と一緒に、データフレームが音声の再構成に使用される符号励振形線形予測復号器に伝送される。

【0020】符号励振形線形予測復号器は、上述した符号化過程と逆の過程を行う。固定確率コードブック指標は、受信器（合成器ともいう）の固定確率コードブックによって受入れられ、同伴ベクトル $e(i)$ （励振信号）がコードブックから選出される。励振信号 $e(i)$ は、逆長時間予測過程（長時間相関が与えられる）を励起するのに使用され、量子化された $x(i)$ の等価情報、山付 $x(i)$ を生じる。再構成音声信号 $y(i)$ は、（短時間相関が与えられる）逆短時間線形予測フィル過程を使用して山付 $x(i)$ をろ波することにより得られる。

【0021】一般的に、再構成励振信号、山付 $x(i)$ は、適応コードブックおよび固定コードブックの基準化された補助情報の和と解釈される。これらのコードブックからベクトルを選出するため、聴覚適合誤差基準が使用される。これは、人間の聴覚系に存在するスペク

$$g(i) = x(i) - \hat{x}(i) + \sum_{n=1}^N \gamma^n a_n g(i-n) \quad (3)$$

【0025】合成による分析符号器の誤差基準は、サブフレーム毎に定式化される。標本 L 個のサブフレーム長について、通常使用される基準は、以下の通りである。 ※

$$E = \sum_{i=\hat{i}}^{\hat{i}+L-1} g(i)^2 \quad (4)$$

【0027】数式中、山付 i は、サブフレームの最初の標本である。この基準は、サブフレーム全体に亘って、励振標本を不等に重み付けする。標本、山付 $x \cdot$ （山付 $i+L-1$ ）は、 g （山付 $i+L-1$ ）のみに作用し、山付 x （山付 i ）は、本サブフレーム中の $g(i)$ の全ての標本に作用する。

【0028】数式（4）の基準は、上記山付 i に先行し★

$$q(i) = \begin{cases} 0, & i < \hat{i}, \\ z(i) - \sum_{n=1}^{\hat{i}-i} \gamma^n a_n q(i-n), & \hat{i} \leq i < \hat{i}+N \\ 0, & i \geq \hat{i}+N \end{cases} \quad (5)$$

【0030】数式中、 $z(i)$ は、本サブフレームに先行して、 $x(i) -$ 山付 $x(i)$ により励振されたときの聴覚重み付け合成フィルタの本サブフレーム内の零入力応答である。

【0031】時間領域においては、因数 γ によるスペクトルデエンファシスは、全極形フィルタのインパルス応答を、より速やかに減衰させることになる。實際上、8 kHz の標本化率と $\gamma=0.8$ について、インパルス応答は、標本 20 個を越えて、エネルギーの有意部を有しない。

【0032】減衰が速いので、全極形フィルタ $1/(1$

* トルマスキングを利用することにより信号と再構成音声信号とを使用する代りに、上記聴覚適合誤差基準は、聴覚重み付け信号の差を判断する。

【0022】信号の聴覚重み付けは、音声中に存在するホルマントに重点をおかない。本範例によれば、全ての極を内側に移動することにより、スペクトルデエンファシスを得ることができる全極型フィルタにより記述される。これは、予測係数 a_1, a_2, \dots, a_N を有するフィルタを係数 $\gamma a_1, \gamma^2 a_2, \dots, \gamma^N a_N$ (γ は、通常、約 0.8 の値に設定される聴覚重み付け因数である) を有するフィルタにより置き換えるのと等価である。

【0023】聴覚重み付け領域での標本化誤差信号 $g(i)$ は、以下の通りである。

【0024】

【数3】

※ 【0026】

【数4】

★て（すなわち、本サブフレームの始まりに先行して） $x(i)$ と上記山付 $x(i)$ との差の効果を含む。本サブフレーム内での励起を定義することにより、重み付けされた合成フィルタの零入力応答を表現することが便利である。

【0029】

【数5】

$-\gamma a_1 z^{-1}, \dots, -\gamma^N a_N z^{-N}$ のインパルス応答は、有限インパルス応答フィルタによって近似しうる。有限インパルス応答フィルタのインパルス応答を h_0, h_1, \dots, h_{R-1} とする。これにより、聴覚重み付けされた音声に作用する誤差基準のベクトル表記法が可能となる。符号器は、サブフレーム毎に動作するので、標本中のサブフレームの長さ L でベクトルを定義することが便利である。例えば、励振信号について、

【0033】

【数6】

$$\hat{x}(i) = \begin{bmatrix} \hat{x}(i) & \hat{x}(i+1) & \cdots & \hat{x}(i+L-1) \end{bmatrix}^T \quad (6)$$

【0034】また、スペクトル重み付け行列Hは、以下
のように定義される。

$$H = \begin{bmatrix} h_0 & 0 & \cdots & \cdots & 0 \\ h_1 & h_0 & \cdots & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdots & \cdot \\ h_{R-1} & h_{R-2} & \cdots & \cdots & \cdot \\ 0 & h_{R-1} & \cdots & \cdots & h_0 \\ \cdot & \cdot & \cdots & \cdots & h_1 \\ \cdot & \cdot & \cdots & \cdots & \cdot \\ \cdot & \cdot & \cdots & h_{R-1} & h_{R-2} \\ 0 & \cdot & \cdots & 0 & h_{R-1} \end{bmatrix} \quad (7)$$

【0036】Hは、元 $(L+R-1) \times L$ を有する。これにより、ベクトル $H \cdot$ 山付 $x(i)$ は、ベクトル、山付 $x(i)$ に対する無限インパルス応答フィルタ $1/(1-\gamma a_1 z^{-1}, \dots, \gamma^R a_R z^{-R})$ の全応答を近似

※する。これらの定義を使用すれば、聴覚重み付けされた最適基準は、以下の通りである。

【0037】

【数8】

$$\varepsilon = \left[x(i) + q(i) - \hat{x}(i) \right]^T H^T H \left[x(i) + q(i) - \hat{x}(i) \right] \quad (8)$$

【0038】Hの現時定義を使用すれば、数式(8)の誤差基準は、自己相関型(HTHは、テープリッツ(Toeplitz)であることに留意してほしい)。行列Hが正方形 $L \times L$ となるように裁断されたときは、数式(8)は、原符号励振形線形予測に使用される、より普通の分散基準である数式(4)に等しくなる。

【0039】符号励振形線形予測符号化の実施例

図1は、符号励振形線形予測符号化に適用された本発明の実施例を示す。デジタル形の音声信号 $s(i)$ は、符号化に供される。信号 $s(i)$ は、線形予測係数 a_n を生成する従来形線形予測分析器100に与えられる。信号 $s(i)$ は、数式(1)により記述される過程に従って動作する従来形線形予測フィルタ(すなわち、「短時間予測器」(STP))120と従来形遅延値推定器140とに供給される。特定の標本時刻において妥当な遅延情報を決定するため、遅延値推定器140は、 $s(i)$ の複数の標本からなる(当特定標本と隣合う中心を有する)窓の従来相関を行う。多数の窓のそれぞれは、同一長さを有する。この相関に関係する窓は、図2に示されている。

【0040】図2は、複数のフレームFと信号 $s(i)$ の複数の標本からなる成分のサブフレームSFとの境界確定を示す($s(i)$ の実標本値は、説明を判り易くするため、省略されている)。3個のフレーム F_{n-1} (過去フレーム)、 F_n (現時フレーム)および F_{n+1} (次フレーム)が示されている。これらのフレームのそれぞれは、信号 $s(i)$ の標本160個からなる。

【0041】フレーム境界の位置決めは、後述する時間軸シフトプロセッサ200により行われる。時間軸シフトプロセッサ200は、原音声信号 $s(i)$ のサブフレームの終端を示す標本位置 d_{p1}' を与える。遅延値推定器140は、フレーム境界に到達する時を知るため、原音声のサブフレーム境界のトラックを保持するだけである(このようなフレーム境界は、整数個のサブフレーム境界に位置する)。遅延値推定器140は、時間軸シフトプロセッサ200が音声フレームに作用する前に該音声フレームに作用するので、遅延値推定器140は、未来フレーム境界の位置を予測しなければならない。遅延値推定器140は、時間軸シフトプロセッサ200により供給される最後フレーム境界にフレーム長(例えば、標本160個)に等しい一定個数の標本を追加することにより、上記予測を行う。

【0042】遅延値推定器140が信号 $s(i)$ の現時フレームと次フレームとの境界 $M(FB_{n+1})$ で妥当な遅延値 M を決定すると仮定する。この決定を行うため、遅延値推定器140は、この境界の両側に存在する、標本160個からなる窓を自らのメモリ内に記憶する(遅延値推定器140は、次フレーム内で妥当な信号 $s(i)$ 標本を受け入れるために待機しなければならない)。この標本窓は、窓Aと名づけられる。ついで、遅延値推定器140は、140個の他の信号 $s(i)$ の窓のうちの最初の窓である窓B1内の信号 $s(i)$ の標本を使用して相関算出を行う。窓B1は、窓Aの始まりよりも標本20個分早く始まり、窓Aの終端よりも標本2

0個分早く終了する標本160個の窓である。窓B1に同伴する相関値は、メモリ内に記憶される。相関過程は、窓B1より標本1個分早く開始される標本160個分の窓である窓B2について繰返される。相関算出は、次の138個の窓のそれぞれについて行われる。各窓は、標本1個分先行する窓から距てられている。

【0043】図2に示されているように、遅延値推定器140は、ほぼ2個の信号標本フレームを記憶するのに十分なメモリを有しなければならない。Dが最長許容遅延値であるときは、メモリは、窓Aの始まりに先行して、標本D個分拡張するべきである。D=160のときは、遅延値推定器140は、 F_{Bn+1} で妥当な推定遅延値を算出するために、フレーム F_{Bn-1} の3番目のサブフレームSF2の始まりからフレーム F_{Bn+1} の2番目のサブフレームSF1の終端までの、信号 $s(i)$ の標本を記憶しなければならない。遅延値Mは、窓Aの全ての標本と最大相関を有する標本窓Bに基づいて遅延値推定器140により決定される。すなわち、遅延値は、最も相関度の高い窓Bが窓Aから時間的にシフトされる標本の数に等しい。遅延値推定器140は、フレーム毎に1回、フレーム境界遅延推定値Mを決定する。また、遅延値推定器140は、フレーム境界において妥当な遅延値の通常線形補間により、各サブフレーム内に入った一定個数の標本(例えば、標本10個)において妥当な遅延値 m を決定する。この目的のため、次フレームへ入った標本10個において必要な遅延値は、フレーム境界における遅延値に等しく設定される。

【0044】遅延値推定器140により供給された遅延値とに同伴するタイミングが図3に示されている。図3に示されているように、フレーム n の両側のフレーム境界において妥当な遅延値は、 $M(F_{Bn})$ および $M(F_{Bn+1})$ である。フレーム n 内の各サブフレーム境界(SB)の後の一定個数の標本において妥当な遅延値は、 $m_n(k)$ 、 $k=0, 1, 2, 3$ として示されている。 $m_n(k)$ のこれらの値は、上述の通り、補間により決定される。遅延値 $m_n(k)$ は、適応コードブックプロセッサ150に供給される。後述するように、適応コードブックプロセッサ150は、この遅延情報を使用することにより、時間軸シフトプロセッサ200へ適応コードブック補助情報を供給する。

【0045】適応コードブックプロセッサ

適応コードブックプロセッサ150は、遅延値推定器140から出力された遅延推定値 $m_n(k)$ と符号励振形線形予測プロセッサから得られた過去再構成音声信号とに基づいて、(符号化しようとする)現時音声サブフレームの推定値を時間軸シフトプロセッサ200に供給する。適応コードブックプロセッサ150は、遅延値 $m_n(k)$ を使用することにより、適応コードブックプロセッサ150のメモリに記憶された過去再構成音声信号に対する遅延ポインタ $d(i)$ を決定するように動作す

る。ついで、選択された過去音声標本、山付 $x(i)$ は、符号化しようとする推定現時音声サブフレームとして時間軸シフトプロセッサ200に供給される。符号化しようとする各原音声サブフレームについて、適応コードブックプロセッサ150は、対応する音声標本サブフレームと次サブフレーム内に入った一定個数の超過標本とを供給する。例えば、この超過標本の一定個数は、10に等しい。

【0046】図4は、適応コードブックプロセッサ150の一実施例を示す。この適応コードブックプロセッサ150は、プロセッサ155とRAM157を含む。プロセッサ155は、過去再構成音声信号、山付 $x(i)$ を受入れ、現時サブフレーム音声標本と次サブフレーム音声標本とを算出するのに使用するため、上記過去再構成音声信号をRAM157に記憶する。プロセッサ155は、上記のような標本値の算出に使用される、遅延値推定器140から出力された遅延値 $m_n(k)$ をも受入れるプロセッサ155は、試原信号の生成に使用するため、上記算出標本値、山付 $x(i)$ を時間軸シフトプロセッサ200に供給する。

【0047】時間軸シフトプロセッサ200に供給された各音声標本値は、以下のように決定される。まず、当標本(すなわち、時間軸シフトプロセッサ200に供給されようとする標本)について妥当な遅延ポインタ $d(i)$ がプロセッサ155により決定される。これは、当標本の両側に存在する(遅延値推定器140により供給された)1対の遅延値 $m_n(k)$ による補間により行われる。遅延ポインタ $d(i)$ を供給するためにプロセッサ155により使用される補間法は、供給された遅延値 $m_n(k)$ による通常線形補間である。ついで、プロセッサ155は、(当標本について妥当な)遅延ポインタ $d(i)$ を、当標本の値として現時フレーム内で使用されるべき先の音声標本の後向きポインタとして使用する。このような先の標本は、RAM157に記憶されている。一般的に、遅延ポインタ $d(i)$ は、過去標本そのものを指し示さず、連続する過去標本間のどこかを指し示しがちである。このような状況では、プロセッサ155は、過去標本を補間することにより、遅延ポインタが関係する時点で妥当な過去標本値を決定する。過去標本値を決定するのにプロセッサ155により使用される補間技術は、ラビナー(Rabiner)とシェイファー(Schafer)とによりのべ述べられた「音声信号のデジタル処理」、26~31頁(1978)のような通常帯域限定補間法である。プロセッサ155によって実現される補間は、遅延値によって示される時刻に最も近い過去標本のいずれの側にも20個のタップを使用する。

【0048】図5の(a)~(c)は、適応コードブックプロセッサ150が現時サブフレーム(および次フレーム)内で使用するために、過去標本を選択する過程を示す。図示を分かり易くするため、図5の(a)~(c)

13

は、算出値 $d(i)$ が過去値間の一点ではなく、過去標本値そのものを指し示すと仮定する。また、遅延値がサブフレーム長より短いことが一般性を損うことなく仮定される。

【0049】図5(a)に示されているように、時間軸シフトプロセッサ200に供給されようとする標本には、現時サブフレーム内の標本と次サブフレーム内の一定個数の標本とが含まれる。プロセッサ155は、現時サブフレームの遅延値 m_{curr} を遅延値推定器140から受取り、先のサブフレームの遅延値 m_{prev} を自らのメモリRAM157に記憶している。 m_{curr} が妥当となる時点より前に位置決めされた現時サブフレームの各標本の値である山付 $x(i)$ を決定するため、プロセッサ155は、当標本の標本時刻 i において妥当な遅延ポイント $d(i)$ を決定する。これは、遅延値 m_{curr} と遅延値推定器140から受入れた最新遅延値 m_{prev} とを使用することにより、標本が妥当となる時点を線形補間することにより行われる。この遅延ポイント $d(i)$ が決定された後、プロセッサ155は、自らのRAM157内の標本の帯域幅限定補間により、当標本より標本 $d(i)$ 個前の時点において妥当な標本値である山付 $x(i-d(i))$ を算出する。ついで、この標本値は、現時サブフレームの当標本のための指定記憶場所に挿入される。

【0050】図5の範例によれば、サブフレーム長は、遅延値より長い。現時サブフレーム内の与えられた標本が決定される過程は、遅延ポイントを決定し、上記与えられた標本として使用しようとする標本値を時間的に遡って探すことに基づく。このように、再構成音声セグメントは、現時サブフレーム内での帯域幅限定補間を使用することにより、ほぼ再生される。したがって、例えば、図5(b)において、与えられた標本、山付 $x(i)$ は、その値を、時間的に遅延値 $d(i)$ だけ先行する、先に決定された標本、山付 $x(i-d(i))$ から得る。この遅延値は、現時サブフレームからの遅延値 m_{curr} と次サブフレームからの遅延値 m_{next} とは、標本、山付 $x(i)$ の両側に存在するので、補間された遅延値がこれらの遅延値であることを除いて、上記の通り決定される。遅延値がサブフレーム長より短いときに、一定利得での信号セグメントの再生は、適応コードブック手法を長時間予測器ろ波手法から区別するものである。

【0051】図5(c)に示されているように、次フレーム内の超過標本は、図5(b)内の超過標本と同一方法で決定される。この場合、現時サブフレームから得られた標本は、次サブフレーム内の標本の値を与えるのに使用される。

【0052】実際には、適応コードブックプロセッサ150の上記処理手順は、まず、現時サブフレームと当次サブフレームの一部の全ての標本時刻の全ての遅延ポイント値 $d(i)$ を算出することにより実現される。ついで、

14

で、標本値を必要とする現時サブフレームまたは次サブフレームの各標本時刻 i について、 $d(i)$ は、標本が「配置」された過去時刻 $i-d(i)$ への参照記号として使用される。一般的に、時刻 $i-d(i)$ に配置された標本は、存在しない。したがって、時刻 $i-d(i)$ の両側に存在する標本の帯域限定補間が必要となる。時刻 $i-d(i)$ に標本を生成する帯域限定補間が行われると、この標本値は、時刻 i に割当てられる。この過程は、必要に応じて、現時サブフレームまたは次サブフレーム内の各標本の回帰過程において繰返される。

【0053】適応コードブックプロセッサ150が現時サブフレームと次サブフレームの一定部分とにおいて使用するための標本を決定すると、これらの標本は、符号励振形線形予測符号化過程において使用する時間軸シフト済原信号を決定する基準として使用するために、時間軸シフトプロセッサ200へ供給される。時間軸シフトプロセッサへ供給された標本は、符号励振形線形予測符号化の合成による分析過程に対する適応コードブック補助情報といわれる。

【0054】長時間予測器の実現適応コードブックの代りに、全極型フィルタが使用されることを理解すべきである。しかし、適応コードブックの実現は、ここに説明するように、特に、遅延値が総じてサブフレーム長より短い状況に適合する。これは、適応コードブックの実現が現時サブフレーム内の長時間予測器補助情報を供給するためにのみ長時間予測器利得（ここでは、コードブック利得）の決定値を必要とすることがないからである。この利得は、後に決定しうる。適応コードブックの場合と違って、長時間予測器の実現全極型フィルタは、遅延値がサブフレーム長より短いときのフィルタ利得値を得る非線形等式の解を必要とする。

【0055】時間軸シフトプロセッサ

時間軸シフトプロセッサ200は、原音声信号が符号化のために常に使用されたとした場合よりも小さい誤差で（符号励振形線形予測のような合成による分析符号化法により）符号化されるように、原音声信号セグメントをシフトする方法を決定する。原音声信号の時間軸シフトのために、時間軸シフトプロセッサ200は、まず、原音声信号内で原音声信号エネルギーの極大値を識別する。上記実施例によれば、時間軸シフトプロセッサ200は、原音声信号の複数の重なり合ったセグメントを選択する。これらセグメントのそれぞれは、識別された極大信号エネルギーを有する。時間軸シフトプロセッサ200は、選択された各セグメントを（適応コードブックプロセッサ150により供給された）適応コードブック補助情報セグメントと比較する。この比較が行われることにより、適応コードブック補助情報のセグメントに最も近似する原音声信号セグメントが決定される。適応コードブック補助情報のセグメントに最も近似した原音声信号のセグメントが決定されると、この原音声セグメント

は、符号励振形線形予測法による符号化のための時間軸シフト済原音声信号の形成に使用される。

【0056】図1に示されているように、時間軸シフトプロセッサ200は、短時間予測フィルタ120から原残差音声信号 $x(i)$ を受入れ、符号励振形線形予測符号化法に使用する時間軸シフト済プロセッサ残差音声信号、山付 $x(i)$ を供給する。図6に示されているように、時間軸シフトプロセッサ200は、プロセッサ210、通常型緩衝メモリ220、230および240、プロセッサ210のプログラムを格納するため通常型ROM250、および、プロセッサ210の結果を格納する通常型RAM260からなる。

【0057】時間軸シフトプロセッサ200の動作を、音声信号に対する時間軸シフトプロセッサ200の動作の開始点を示す図7とプロセッサ210の動作のフローチャートを示す図8および図9とを参照して説明する。

【0058】図7に示されているように、時間軸シフトプロセッサ200は、適応コードブックプロセッサ150から出力された適応コードブック補助情報を表わす、再構成音声バッファ220を受入れた後、動作し始める。上述した通り、この適応コードブック補助情報は、適応コードブックプロセッサ150により、現時サブフレームと次サブフレームの一定部分との中へ写像された過去再構成音声標本(図5および関連記述参照)とからなる。この再構成音声バッファは、プロセッサ210による使用のため、RAM260内へロードされる。ポイント $d p 1$ は、適応コードブックと固定確率コードブック補助情報との両方が決定される最後サブフレームの終端を示すため、プロセッサ210により保持され、RAM260に記憶される。上記サブフレームの長さ $subframe-1$ は、一定であり、メモリ(例えば、ROM250)内に保持される。プロセッサ210の先の動作に基づいて、時間軸シフト残差、波付 $x(i)$ がポイント $d p m$ により特定された時点までに、新規作成される(ポイント $d p m$ は、常に、ポイント $d p 1$ 以上である)。また、現時サブフレームに同伴する原残差信号を含む、原残差信号 $x(i)$ の一部は、緩衝記憶装置230に入力され、RAM260内に記憶される。プロセッサ210は、時間軸シフト済信号内の最新標本と原残差音声信号内の対応する標本との標本ずれ(すなわち、累積シフト)を表わす値 $acc-shift$ (RAM260内に)保持する。初期化により、上記状態は、 $d p m = d p 1$ および、 $acc-shift = 0$ を含むように修正される。

【0059】この1組の条件が与えられると、時間軸シフトプロセッサ200は、適応コードブック補助情報に最も整合する、現時サブフレームの時間軸シフト済残差信号(事情により、次サブフレームの一部)を決定するように動作する。

【0060】図8および図9は、図6に示されたプロセッサ210の動作を示すフローチャートを示す。図8に

よれば、プロセッサ210により行われる最初のタスクは、時間軸シフト済残差、波付 $x(i)$ が現時サブフレームの終端まで達するか、または、この終端の後まで伸びているかを決定することである。図7に示されているように、時間軸シフト済残差が拡大した範囲は、ポイント $d p m$ により与えられる。現時サブフレームの終端は、現時サブフレームポイント $d p 1$ と固定サブフレーム長 $subframe-1$ との和により示される。 $d p m < d p 1 + subframe-1$ のときは、時間軸シフト済残差を拡大するために、追加処理が行われる。上記条件が成立しないときは、現時サブフレームについて追加のシフト処理が必要とされない(ステップ305参照)。追加のシフト処理が必要なときは、プロセッサ210は、原残差音声信号 $x(i)$ のセグメント内の最大エネルギーを決定する。通常、最大エネルギー位置は、有声音のピッチパルスの位置に対応する。しかし、これは、必ずしも、事実ではない。最大エネルギーがピッチパルスまたは信号の他の特徴(例えば、エネルギーノイズ)に関連するか否かにかかわらず、最大エネルギー位置の探索は、原信号のシフトが原音声内のエネルギーの特徴を適応コードブック補助情報の有意特徴と最も一致するように、行われる。

【0061】探索されようとする原残差音声信号のセグメントの始まりは、原残差音声信号標本のポイントに関連して定義される。この標本は、シフト済残差信号内のポイント $d p m$ により特定された標本に対応する。この残差音声信号標本ポイント $d p m'$ は、標本ポイント $d p m$ と、波付 $x(i)$ および $x(i)$ 間の累積シフトとの和として決定され、 $d p m' = d p m + acc-shift$ となる(ステップ310参照)。ついで、ポイント $offset$ により指定されて、探索されようとする時間間隔の始まりが算出される(ステップ315参照)。ついで、探索されようとする間隔の長さが決定される(ステップ320参照)。

【0062】ついで、信号 $x(i)$ のセグメント内の最大エネルギー位置が決定される(ステップ325参照)。この決定は、標本5個長窓を使用することにより行われる。原残差音声信号の i 番目の標本と隣合う中心を有する上記標本5個長窓は、エネルギー算出に使用される原残差音声信号の標本を決定する。標本位置 i におけるエネルギーは、窓内の全ての標本の自乗の和によって決定される。 $(i+1)$ 番目の標本位置におけるエネルギーも同一方法で決定される。しかし、窓は、窓の中心位置が $(i+1)$ 番目の標本を含むように、標本1個分、時間的に後の方へ移動される。再び、エネルギーは、窓内の全ての標本値の自乗の和として決定される。セグメント内の各標本位置のエネルギーも同一方法で決定される。現時窓内の標本のエネルギーは、直近の過去標本窓のエネルギー窓外へシフト済標本のエネルギー+窓内へシフト済標本のエネルギーとして決定されうる。この方法で決定された最大エネルギーを同伴する標本位置は、ポイント $location$ によ

り識別される。

【0063】原残差信号 $x(i)$ のセグメントがセグメント内に最大エネルギーを有する標本について探索されると、プロセッサ210は、この最大エネルギー標本が先のサブフレーム内で判断された標本であるか否かを決定する（したがって、最大関心事出はない）。これは、locationが dpm' に先行するか否かを決定することにより行われる（ステップ330参照）。

【0064】locationが dpm' に先行するときは、他の探索がプロセッサ210により行われる。しかし、この場合、探索されたセグメントは、offset（相対位置） $=location+0.75$ 遅延値（ステップ335参照）として指定される標本から始まり、 0.5 遅延値分持続する。遅延値は、遅延値推定器140により、現時サブフレームM（F_{Bn}）の始まり時に妥当な遅延値として与えられる。原残差信号内の諸有意ピッチパルスエネルギー特徴は、多分、一遅延値分距てられるので、新しいoffsetの算出により、探索が前方へ（ 0.75 遅延値）飛越すことができるようになり、多分、 0.5 遅延値の長さのセグメント内での最大エネルギー特徴を見付けられる。最大エネルギーを有する標本位置は、ステップ325に関係して上述した通りに決定される（ステップ345参照）。

【0065】locationが dpm' に先行しないときは、多分、 dpm' 後の最初のピッチパルスが見付けられているので、制御の流れは、ステップ350に飛ぶ。

【0066】ステップ325またはステップ345において決定された最大信号エネルギーの位置が $dpm'+$ 遅延値となる時は、（確実ではないが）多分、 dpm' の後であって、 $dpm'+$ 遅延値より前に位置するピッチパルスは、プロセッサ210によりこの時刻まで行われた探索により、検出されていない（ステップ350）。この場合、原残差信号の他のセグメントが定められ、このセグメント内で最大エネルギー位置が決定される。ステップ325または345で決定された最大信号エネルギー位置が $dpm'+$ 遅延値に先行するときは、制御の流れは、ステップ380に飛ぶ。ステップ350が原残差音声信号の他のセグメントを探索しなければならない結果になったと仮定すれば、このセグメントは、 $offset=location-1-2.5$ 遅延値（ステップ355参照）から始まり、長さ $=0.5$ 遅延値分前向きシフトする（ステップ360参照）ように決定される。最大エネルギー位置は、ステップ325に関係して上述した通りに決定される。ただし、この位置の標本ポイントは、location 2に記憶される（ステップ365参照）。

【0067】最大エネルギー位置（location 2）が dpm' より後のときは、location 2は、 dpm' より後の最初のピッチパルスの位置を識別し、locationは、location 2に等しくなるように設定される（ステップ370および375参照）。他方、最大エネルギー位置が dpm'

m' の後でないときは、location 2は、 dpm' より後の最初のピッチパルスでなく、locationは、ステップ325または345において割当てられた値に設定されたままである（このような状況においては、ポイントであるlocationは、ステップ365での動作により書きされない）。

【0068】この時点で、原残差信号のセグメント内の最初のピッチパルス（すなわち、最大エネルギー）位置が見付けられる。今度は、この位置を含む原残差信号のセグメントが信号内の標本にある一定のポイントを設定することにより、プロセッサ210により定められる。これらのポイントは、決定されたlocationを含む上記セグメントの始まり（sfstart）と終端（sfend）とを指定する。このセグメントは、後に使用するため、適応コードブック補助情報に最も整合するように原残差信号を調整（すなわち、シフト）する過程の一部として定義される。

【0069】まず、セグメントポイントの省略時値は、プロセッサ210により設定される。ポイントsfstartは、 dpm' に等しくなるように設定されるので、標本位置は、 $dpm+acc-shoft$ に対応する（ステップ380参照）。sfstartのこの値は、 $x(i)$ と0となる波付 $x(i)$ との間の追加累積シフトに対応する。すなわち、 dpm' （ $=sfstart$ ）から始まる、 $x(i)$ の一部を使用しても、原信号とシフト済残差信号との間の累積シフトは、まったく増加しない。

【0070】ポイントsfendは、 $location+extra$ に設定される。extra値は、メモリ（例えば、ROM250）内に記憶された定数であり、標本の一定数（例えば、標本10個）に等しい。extraを使用すれば、原残差音声のピッチパルス（すなわち、最大エネルギー）がこれらのポイントにより識別された原残差音声のセグメントの終端に来ることのないように保証される（ステップ380参照）。

【0071】ポイントsfendの省略時値は、一定の状況の下で、書きされうる。sfendの省略時値が、原残差音声のセグメントが適応コードブック補助情報の終端を越えて有意に前向きにシフトすることを意味するときは、ポイントsfendは、 $dpm'+subframe-1+extra$ （ただし、subframe-1は、上述した通り、固定適応コードブックサブフレーム内の標本の数に等しい定数である）で終わるように設定される（ステップ385および390参照）。

【0072】識別されたピッチパルス（すなわち、主エネルギー）の位置が適応コードブックのサブフレームの終端を有意に越えるときは、sfendの値は、さらに、書きされうる。このような状況の下では、セグメントは、適応コードブックサブフレーム境界の終端において終ると判断される（ステップ395および400参照）。このようなsfendの定義は、ピッチパルス（すなわち、主

エネルギー)位置がセグメントの終端より後になることを意味する。したがって、セグメントは、もはや、ピッチパルスを含まない。

【0073】この時点で、識別されたピッチパルス(すなわち、最大エネルギー)の位置は、 $sfstart$ で始まり、 $sfend-1$ で終る標本範囲の外に来るか否かを決定するために、検査される(ステップ405参照)。識別されたピッチパルスの位置が上記標本範囲の外に来るときは、波付 $x(i)$ は、 $acc-shift$ を変更することなく、 $x(i)$ の帯域限定補間により得られた標本で伸ばしうる(すなわち、制御の流れは、ステップ480に飛びうる)。その他のときは、シフトが行われる(ステップ410~475参照)。

【0074】識別されたピッチパルス(すなわち、主エネルギー)の位置が上記範囲の外にないときは、適応コードブック補助情報のL長部(dpm から始まり、 $dpm+L$ で終る)に最も密接に整合する、 $x(i)$ の標本($sfstart$ および $sfend$ により区切られたセグメントに隣合う指定標本範囲内の) L個の集合(すなわち、セグメント)がプロセッサ210により決定される。

【0075】 $x(i)$ の上記L長セグメントは、 $x(i)$ の与えられたL長セグメントが山付 $x(i)$ のL長部に整合する程度によって、 $sfstart$ と $sfend$ とにより区切られた $x(i)$ のセグメントの上記L個の標本からなり、または、 $sfstart$ と $sfend$ とに関してシフトされたセグメントの(帯域限定補間により得られた)標本からなる。この決定に関する述語として、可能標本シフトの範囲の制限(ステップ410参照)と標本長 L とが決定される(ステップ415参照)。 $x(i)$ のL長セグメントと適応コードブック補助情報、山付 $x(i)$ との「密接度」(すなわち、類似度)の決定は、これらの信号の相互相関過程を通じて行われる(差信号または誤差信号のような他の類似度も使用しうることも理解されよう)。山付 $x(i)$ のセグメントとの相互相関に使用するための $x(i)$ のL長セグメントの選択は、図10に関して説明する。

【0076】図10は、ステップ310~400に関係して上記した通りに配置された原残差音声信号 $x(i)$ のセグメントの一例を示す。セグメントは、標本 $sfstart$ から始まり、標本 $sfend$ で終る。ピッチパルスは、標本 $location$ に存在する。標本 $location$ と標本 $sfend$ との距離は、 $extra$ に等しい。上述した通り、ポイント $sfstart$ とポイント $sfend$ とにより区切られたセグメント内に入る $x(i)$ の標本は、零シフトに対応する。 $x(i)$ のシフト済セグメントは、この零シフト位置との関係で定義される。各シフト済セグメントは、長さ L であり、零シフト位置との関係で一定の正数または負数の標本長(または、小数の標本長)から始まる(および終る)換言すれば、各シフト済セグメントは、 $sfstart+shift$ で始まり、 $sfend+shift$ で終る。図10に示されてい

るように、 $shift$ の可能シフト値の範囲は、 $\pm limit$ である。

【0077】したがって、例えば、一の可能シフト値は、 $shift=-limit$ となる。この場合、このようなシフトで定義された $x(i)$ のL長セグメントは、位置 $sfstart-limit$ で始まり、位置 $sfend-limit$ で終る。同様に、他の一の可能シフト値は、 $shift=+limit$ となる。この場合、このようなシフトで定義された $x(i)$ のL長セグメントは、位置 $sfstart+limit$ で始まり、位置 $sfend+limit$ で終る。上述した通り、 $\pm limit$ は、可能シフト範囲を指定する。したがって、 $sstep$ の大きさのシフト幅寸法(すなわち、シフト精度)が与えられれば、 $shift$ は、範囲($-limit \leq shift \leq +limit$)内の値を取りうる。シフト幅寸法 $sstep$ は、一例として、標本0、5個に設定されうる。小数で表わされるシフトから生じる標本値は、通常の帯域限定補間により決定される。原残差信号 $x(i)$ の複数の $2 \times limit / sstep$ セグメントは、この方法で定義されうる。全てが $\pm limit$ 内のL長セグメントである。ただし、各セグメントは、その近接セグメントと重なり合うとともに、最近接セグメントから標本 $sstep/2$ 分ずれている。

【0078】 $limit$ と $extra$ の相対寸法は、システム性能に影響する。例えば、 $extra$ が小さくなるにつれて、符号化遅延値は、小さくなる。しかし、 $shift$ が $x(i)$ のL長セグメントからピッチパルスを除外する値を取る確率が高くなる。この除外が生じたときは、音声信号中に可聴歪みが生じる。また、除外確率は、 $limit$ が大きくなるにつれて高くなる。除外が生じないように保証するのを助けるため、 $limit$ の値は、 $extra$ の値より小さくすべきである。例えば、 $extra$ の値が10のときは、 $limit$ の値は、6に設定されうる。

【0079】上記の通り識別された、 $x(i)$ の上記L長セグメントのそれぞれについて、セグメントと適応コードブック補助情報のL長セグメント、山付 $x(i)$ との類似度が算出される。この算出は、一例として、相互相関である。各相互相関に使用される適応コードブックセグメントは、 dpm で始まり、 $dpm+L$ で終る(図7参照)。相互相関は、 $sstep$ に等しいシフト幅寸法で行われる($sstep$ が非整数値に等しいときは、 $x(i)$ および山付 $x(i)$ の全てのセグメントについて必要標本値を供給するため、山付 $x(i)$ の通常、帯域限定補間が先行して行われる)。各相互相関は、相互相関値(すなわち、類似度)を生じる。このような全ての相互相関は、 $sstep$ だけ時間的に距てられた相互相関値集合を形成する。したがって、この集合の各相互相関値は、この値の算出に使用される $x(i)$ のL長セグメントに対応する $shift$ と関連する。

【0080】相互相関値集合が決定されると、適応コードブックセグメントと最大相互相関を有する、原残差信号のセグメントが増大した時間分解能により決定される

21

(ステップ450参照)。一例として、これは、3個の連続相互相関値からなる各集合について2次多項式曲線を決定する(3個の値からなる集合は、その最近接集合から1個の値だけ、ずれている)。集合中のこれら3個の相互相関値の中央値は、上記の通り、シフト済原残差信号に対応する。3個の相互相関値からなる集合(したがって、同伴多項式曲線)は、上記中央値とその同伴シフトとにより識別される。このような各曲線について、最大値とこの最大値位置(loc-max)とが決定される。最大値位置loc-maxが上記3個の値からなる範囲の外に存在するときは、これら3個の値と同伴曲線とは、無視される。最も大きい最大値を有する曲線は、適応コードブック補助情報のセグメントとの最良整合を生じる原残差信号のシフトを識別する。

【0081】最良整合を生じる原残差信号シフトは、最も大きな最大値を有する多項式曲線の最大値の位置を知ることにより改良される。上記曲線に同伴する、3個の相互相関値の中央値の位置に関して定められた最大値を使用すれば、shiftは、 $\text{shift} = \text{shift} + \text{step} * \text{loc-max}$ と改良されうる。

【0082】この時点で、原残差信号の最良シフトが決定されている。ついで、このシフトは、シフト済残差信号、波付x(i)を持続時間Lだけ前向きシフトさせるのに使用されうる。このシフトは既知であるから、原残差信号x(i)とシフト済残差信号、波付x(i)との間の累積シフトは、 $\text{acc-shift} = \text{acc-shift} + \text{shift}$ として更新される(ステップ475参照)。

【0083】累積シフトが更新されると、シフト済残差信号x(i)は、acc-shiftを、shiftに対応する、原残差信号のセグメントの使用に整合させるために、前向きシフトされる。原残差標本値は、原信号標本時刻においてのみ、入手可能であることに留意してほしい。しかし、原残差信号の最適シフト決定によればアップサンプリングが相互相関の算出前に行われており、(一般的に、非整数である)値loc-maxが決定されている。一般的に、これは、シフト済残差信号、波付x(i)とシフ*

$$\lambda(i) = \frac{\bar{x}(i)^T \hat{x}(i)}{\bar{x}(i)^T \bar{x}(i)}$$

【0089】数式中、波付x(i)および山付x(i)は、サブフレームと等しい長さの信号である。この基準化因数は、山付x(i)に乘じられ、プロセッサ200から出力される。

【0090】再び図1を参照すれば分るように、波付x(i)と適応コードブック推定値λ(i)・山付x(i)とは、修正された原信号、波付x(i)から推定値λ(i)・山付x(i)を差引く回路160に供給される。結果は、固定確率コードブック検索プロセッサ170に供給される励振残差信号γ(i)となる。

【0091】コードブック検索プロセッサ170は、因

22

*ト済残差信号を前向きシフトするのに使用されるべき原残差信号x(i)との間の非整数的標本時刻関係を生じる。したがって、シフト済残差信号の標本と時刻一致する原信号の標本値を供給するために、原信号のL長セグメントの帯域限定補間が使用される。このような時刻一致が行われると、この時刻一致済信号の全ての標本は、既存のシフト済信号と連結されうる(ステップ480参照)。

【0084】制御の流れは、累積シフトを更新することなく、ステップ480に飛び越えることができたことに留意してほしい。この場合、標本L個の長さの原信号は、補間されることにより、シフト済残差信号の標本に、先行シフト済残差セグメントと同一acc-shift値を供給する。

【0085】いずれの場合にも、dpmは、波付x(i)の前向きシフトを表わすように更新される(ステップ490参照)。

【0086】図8および図9に示されているように、dpmが更新されると、制御の流れは、ステップ305に戻る。上述した通り、ステップ305は、シフト済残差信号が現時サブフレームの終端を越えて前向きシフトするのに、追加処理が必要か否かを決定する。追加処理が必要なときは、シフト済残差の追加の前向きシフトが行われるように、再び、制御は、図8および図9に示されたステップ310~490に示された過程を経る。ステップ310~490は、ステップ305の条件が満されるまで、繰返される。シフト済残差が現時適応コードブックサブフレームの終端まで、または、越えて前向きシフトされると、適応コードブックサブフレームの終端のポイントは、更新され(ステップ500参照)、原残差の時間軸シフトに伴う処理は、終了する。

【0087】波付x(i)が時間軸シフトプロセッサ200により決定されると、基準化因数λ(i)がプロセッサ210により、以下の通り決定される。

【0088】

【数13】

(13)

数μ(i)により基準化された全ての固定確率コードブックベクトルz(i)のうち、いずれが最小2乗され聴覚重み付けされた意味で、最も密接にr(i)に整合するかを決定するように、通常通り動作する。選定され基準化された固定コードブックベクトルμ(i)zmin(i)は、基準化された適応コードブックベクトルλ(i)・山付x(i)に追加されることにより、現時再構成音声信号の最良推定値、山付x(i)を生じる。この最良推定値、山付x(i)は、適応コードブックプロセッサ150により、そのメモリ内に記憶される。

【0092】通常音声符号器の場合のように、適応コー

ドブック遅延値および基準化因数入およびM、固定確率コードブック指標IFC、および、利得 $\mu(i)$ および線形予測係数 a_n は、通常符号励振形線形予測復号器(受信器)により、再構成用チャネルを通じて伝送される(図13参照)。この伝送は、上記パラメータをもたらす信号の形を有する。本発明の実施例の作用により(符号化過程において)誤差が少なくなるので、サブフレーム毎に1回でなく、フレーム毎に1回適応コードブック遅延情報を伝送することが可能である。サブフレーム遅延値は、発信器の遅延値推定器140により行われたのと全く同一の方法で遅延値を補間することにより、受信器側で提供しうる。

【0093】サブフレーム毎でなくフレーム毎に適応コードブック遅延情報を発信することにより、遅延に伴う必要帯域幅は、有意に削減しうる。

【0094】図9に示されたステップ475に関連して上述したように、acc-shiftは、原信号 $x(i)$ とシフト済信号、波付 $x(i)$ との間の時間的累積シフトを表わす。これらの信号の常に増大する非同期性を防止するため、遅延値推定器140は、時間についてMの計算値を調整しうる。遅延値推定器140により達成されるこの目的に適した調整過程は、図12に関連して説明する。

【0095】図12は、状態A、BおよびCを有する有限状態機械を示す。この機械の状態は、常に増大する非同期性を防止するために、Mの計算値に対する調整量を表わす。状態間の遷移は、時間軸シフトプロセッサ200により供給されるacc-shiftの値に基づく。機械が状態Aの時は、目標遅延値 $m_n(k)$ を決定するのに使用される遅延値 $M(FB_{n+1})$ は、調整されない。状態B

【0096】

【数14】

$$M(FB_{n+1}) = M(FB_n) + \delta$$

【0097】数式中、 δ は、説明のため、1標本時間に等しい。状態Cのときは、機械は、 $M(FB_{n+1})$ を以下の通り調整する。

【0098】

【数15】

$$M(FB_{n+1}) = M(FB_n) - \delta$$

【0099】初期状態(A、BまたはC)が与えられると、有限状態機械は、acc-shiftの値のトラックを保持することにより、動作する。acc-shiftの値が現時状態と他の状態との間の遷移条件を満たすものであるときは、上記他の状態への遷移が生じる。例えば、機械が(遅延値推定器140の説明のための初期状態である)状態Aにあり、 $-3ms < acc-shift < 3ms$ であると仮定すれば、機械は、状態Aのままであり、 $M(FB_{n+1})$

は、修正されない。acc-shiftの値が3msより大きいときは、機械は、状態Cへ遷移し、 $M(FB_{n+1})$ は、acc-shiftにより示された非同期性を相殺するのを助けるため、1標本時間だけ増加される。他方、状態Aのときに、acc-shiftが-3msより小さくなるときは、機械は、状態Bへ遷移し、 $M(FB_{n+1})$ は、非同期性を相殺するのを助けるため、1標本文減少される。この動作は、状態BおよびCについても同様である。

【0100】他の実施例

図1に示された実施例に対する追加の実施例が図11に示されている。この実施例によれば、試原信号生成器610は、原ディジタル音声信号 $x(i)$ を受入れ、複数の試原信号、波付 $x(i)$ を生成する。試原信号生成器610は、図1、6、8および9に示されたものと類似の時間軸シフトプロセッサからなる。しかし、この時間軸シフトプロセッサは、試原信号と適応コードブック補助情報との相関を行わない。すなわち、この時間軸シフトプロセッサは、原音声信号 $x(i)$ の複数回のシフトに基づいて複数の長試原信号を供給するのみである。図10に関連して上述したように、これらの試原信号は、標本sfstartから始まり標本sfendで終る長セグメントに関係して、 $\pm limit$ の範囲に亘ってシフト幅寸法sstepのシフトにより決定された、原信号の長セグメントである。試原信号生成器610は、原残差信号と試原信号との相互相関を行わないので、自らは、符号化のための試原信号を選択しない。試原信号生成器610は、試原信号を供給するよりも、むしろ、処理のために符号(すなわち、合成)器620に対して生成する。

【0101】符号器620は、図14に示された符号励振形線形予測のような従来型合成による分析符号器からなる。合成(すなわち、再構成)原信号、山付 $x(i)$ は、適応コードブック出力信号と固定コードブック出力信号の和 $e(i) + \lambda(i)x(i-d(i))$ として図14に示された信号である(図14に示された回路45参照)。符号励振形線形予測符号の分析処理により決定された、(合成信号、山付 $x(i)$ が生成される)符号化された信号パラメータは、後の使用のため、RAMに記憶されうる。したがって、符号器620の出力、山付 $x(i)$ は、与えられた試原信号、波付 $x(i)$ に基づく原信号 $x(i)$ の推定値である。ついで、この、原信号の推定値は、推定原信号、山付 $x(i)$ と試原信号、波付 $x(i)$ との類似度を決定するため、試原信号と比較される。この類似度は、2個の信号間の差(すなわち、誤差)信号 $E(i)$ を決定する差回路630に供給される。誤差信号 $E(i)$ は、与えられた試原信号に同伴する誤差のトラックを保持する試原信号生成器610に供給される。すべての試原信号が上記の方法で処理されると、試原信号生成器610は、いずれの試原信号、波付 $x(i)$ が最高類似度(例えば、最小誤差)を生じたかを決定しうる。ついで、試原信号生成器610

は、符号器620に信号を供給することにより、最小誤差を有する試原信号に同伴する、記憶された符号パラメータを使用させる。これらのパラメータは、原信号x(i)の符号化表現として受信器に伝送される。

【0102】当業者であれば、「原」信号、「再構成」信号等の信号について言われたことは、これらの信号のセグメントにも言うことを理解するであろう。また、与えられた信号がアップサンプリングされるか否かは、その「原」信号、「試原」信号等としての性格を変えない。したがって、例えば、「原信号」に関して「標本」の語を使用することは、アップサンプリングにより供給された信号の標本値、アップサンプリングの結果でない標本、または、この両方を含むうる。

*【0103】表の説明

上記第1実施例に係るソフトウェアプログラム集合は、明細書の一部を成す。この集合のすべてのソフトウェアプログラムは、「C」プログラミング言語で書かれている。本発明の具体化は、汎用コンピュータ（例えば、シリコン グラフィックス社から販売されているアイリスインディゴという名のワークステーション上で上記プログラムを実行することにより、行いうる。サブルーチン「cshiftframe」および「modifyorig」は、全体として、図8および図9に示された関数に対応する。

【0104】

【表1】

Nov 16 09:07 1992 mod.c Page 1

```
#include "macro.h"
/*
 * mod - modify residual
 */
void mod( residualm, accshift, d_shift, shiftr, exctation, residual,
          dpl, dpm, lpcw, lpcorder, delay, subframe1, extra, fcnt)
float *residualm; /* output: modified residual signal */
float *accshift; /* output: shift from mresidual to residual */
float *d_shift; /* output: local shift for all samples */
float shiftr; /* input: maximum shift range */
float *exctation; /* input: adaptive codebook excitation */
float *residual; /* input: original residual */
int dpl; /* input: pointer to output signals */
int *dpm; /* in/out: pointer to end of residualm */
float *lpcw; /* input: weighted lpc coefficients */
int lpcorder; /* input: lpc order */
float delay; /* input: delay */
int subframe1; /* input: subframe length */
int extra; /* input: additional exctation constructed */
long fcnt;
{
    void cshiftframe();
    void modifyorig();
    float shiftr2;
    int sfstart, sfend;

    while( *dpm < dpl+subframe1){
        cshiftframe( &sfstart, &sfend, &shiftr2, *dpm, residual, dpl,
                     *accshift, shiftr, delay, subframe1, extra, fcnt);
        modifyorig( residualm, accshift, d_shift, dpm, shiftr2, exctation,
                     residual, sfstart, sfend);
    }
}
```

【0105】

【表2】

Nov 18 14:56 1992 cshiftframe.c Page 1

```

#include "macro.h"
/*
 * cshiftframe - find optimal frame shift
 */
void cshiftframe( sfstart, sfend, maxshift2, dpm, residual, dpl, accshift,
                  maxshift, delay, subframe1, extra, fcnt)
int *sfstart;      /* output: shift-frame start */
int *sfend;        /* output: shift-frame ending */
float *maxshift2;  /* output: one-sided shift range */
int dpm;           /* output: up to where residualm exists */
float *residual;   /* input : original residual signal */
int dpl;           /* input : output signal pointer */
float accshift;    /* input : shift of output versus input */
float maxshift;    /* input : maximum shift range */
float delay;       /* input : local pitch value */
int subframe1;     /* input : subframe length */
int extra;         /* input : additional excitation beyond current frame */
long fcnt;         /* input : frame counter (DEBUG) */
{
    void maxeloc(); /* determine location of max energy */
    float maxener;
    int offset;
    int iacshift;
    int length;
    int loc, loc2;

    if( delay < 0 ){
        iacshift = -accshift + 0.5;
        iacshift = -iacshift;
    }
    else
        iacshift = -accshift + 0.5;

    /* determine first a pitch pulse somewhere near dpm */
    length = 1.5 * delay;
    offset = dpm + iacshift - 0.25 * delay;
    maxeloc( &loc, &maxener, residual, offset, length, 2);
    loc -= iacshift;
    printf("cshiftframe: firstloc %d ", loc - dpl);

    /* now find the first pitch pulse for sure */
    if( loc < dpm ){
        offset = loc + iacshift + 0.75 * delay + 0.5;
        length = 0.5 * delay;
        maxeloc( &loc, &maxener, residual, offset, length, 2);
        loc -= iacshift;
        printf(" Aloc %d", loc - dpl);
    }
    if( loc > dpm+delay ){
        offset = loc + iacshift - 1.25 * delay + 0.5;
        length = 0.5 * delay;
        maxeloc( &loc2, &maxener, residual, offset, length, 2);
        loc2 -= iacshift;
        if( loc2 >= dpm ) loc = loc2;
        printf(" Bloc %d", loc - dpl);
    }
}

```

【0106】

【表3】

Nov 18 14:56 1992 cshifcframe.c Page 2

```

*sfstart = dpm;
*sfend = loc + extra;
*maxshift2 = maxshift;
if( *sfend > dpl + subframe1 + extra)
    *sfend = dpl + subframe1 + extra;
if( loc >= dpl + subframe1 + extra/2)
    *sfend = dpl + subframe1;

if( loc >= *sfend || loc < *sfstart)
    *maxshift2 = 0;

printf(" loc is: %d\n", loc-dpl);
/* debugging pictures */
/*
{
    char title[100];
    static float w1[200], w2[200];
    register int i;

    for( i=0; i<200; i++) w1[i] = 0.0;
    w1[loc-dpl-1] = 50.0; w1[loc-dpl+1] = 50.0; w1[loc-dpl] = 100;
    for( i=0; i< subframe1+extra; i++) w2[i] = residual[dpl+iacshift+i];
    for( i=0; i<*sfstart-dpl; i++) w2[i] = 0.0;
    for( i=*sfend-dpl; i<subframe1+extra; i++) w2[i] = 0.0;
    sprintf(title, "shiftrange %5.3f", *maxshift2);
    pictures3( residual+dpl+iacshift, subframe1+extra, w1, subframe1+extra,
               w2, subframe1+extra, fcnt, title, "considered", "shifted");
}
*/

```

[0107]

* * [表4]

Nov 16 09:07 1992 maxeloc.c Page 1

```

#include "macro.h"
void maxeloc( maxloc, maxener, signal, dp, length, ewl)
int *maxloc; /* output: location of maximum energy */
float *maxener; /* output: energy at loc */
float *signal; /* input: signal for which energy is to be found */
int dp; /* input: data pointer into signal */
int length; /* input: window of data */
int ewl; /* input: half length of energy window */
{
    float ener;
    register int i;
    int tail, front;

    ener = 0.0;
    front = dp + ewl;
    tail = dp - ewl;
    for( i=tail; i<=front; i++)
        ener += signal[i] * signal[i];
    *maxloc = dp;
    *maxener = ener;
    for( i=1; i<length; i++){
        front++;
        ener += signal[front] * signal[front] - signal[tail] * signal[tail];
        tail++;
        if( *maxener < ener){
            *maxloc = i + dp;
            *maxener = ener;
        }
    }
}

```

[0108]

[表5]

Nov 18 12:37 1992 modifyorig.c Page 1

```

#include "macro.h"
/*
 * modifyorig - modify original
 */
void modifyorig( residuallm, accshift, d_shift, dpm, shiftrange,
                 exctation, residual, dpl, sfend)
float *residuallm; /* output: modified residual signal */
float *accshift; /* in/out: accumulated shift */
float *d_shift; /* output: local shift value */
int *dpm; /* output: first nonvalid sample of residuallm */
float shiftrange; /* input : one side of shift range */
float *exctation; /* input : excitation waveform */
float *residual; /* input : original residual signal */
int dpl; /* input : window start */
int sfend; /* input : window end */
{
    void bl_interp();
    void getcrit();
    void testi_ubound();
    int k;
    float criterion, best;
    float shift;
    float optshift;
    float locmax;
    int leftlimit, rightlimit;
    int length;
#define MAXDIM 100
    float crit[MAXDIM];
    float a, b;
    float sstep;

    length = sfend - dpl;

    /* first we upsample by a factor 2 */
    sstep = 0.5;
    rightlimit = shiftrange/sstep + 0.5;
    leftlimit = -rightlimit;
    if( leftlimit == rightlimit) rightlimit = leftlimit - 1;
    printf("modifyorig: llim %d rlim %d", leftlimit, rightlimit);
    testi_ubound( rightlimit*2+1, MAXDIM, "modifyorig.c1");
    for(k=leftlimit; k<=rightlimit; k++){
        shift = *accshift + k * sstep;
        getcrit( crit+k-leftlimit, residual+dpl, exctation+dpl, shift, length);
    }

    /* then we interpolate the criterion */
    best = 0.0;
    optshift = *accshift;
    for(k=leftlimit+1; k<rightlimit; k++){
        shift = *accshift + k * sstep;
        a = crit[k-leftlimit+1] + crit[k-leftlimit-1] - 2.0 * crit[k-leftlimit];
        criterion = -2.0;
        if( a != 0.0){
            b = crit[k-leftlimit+1] - crit[k-leftlimit-1];
            locmax = - b / (2.0 * a);
            if( locmax <= 0.5 && locmax >= -0.5)

```

[0109]

40 【表6】

Nov 18 12:37 1992 modifyorig.c Page 2

```

        criterion = a * locmax * locmax + b * locmax + 2.0 * crit[k-leftlimit];
    }
    if( criterion > best){
        optshift = shift + sstep * locmax;
        best = criterion;
    }
    }
    *accshift = optshift;

    printf(" optshift %5.2f best %.4e\n", optshift, best);
    if( best<=1.0){
        for(k=leftlimit+1; k<rightlimit; k++){
            printf("k=%d %f\n", k, crit[k-leftlimit]);
        }
        for( k=0; k<length; k++){
            bl_intrp( residualm+dpl+k, residual+dpl+k, *accshift, 0.9, 8);
            d_shift[dpl+k] = *accshift;
        }
        *dpm = dpl+length;
    }
}

```

[0110]

* * 【表7】

Nov 16 09:07 1992 bl_intrp.c Page 1

```

#include "macro.h"
/*
 * bl_intrp - band-limited interpolation
 */
void bl_intrp( output, input, delay, factor, fl)
float *output; /* output: interpolated output value */
float *input; /* input : array to be interpolated */
float delay; /* input : delay where actual input is */
float factor; /* input : cut-off frequency (relative to fs) */
int fl; /* input : filter length is 2*fl+1 */
{
    /* NOTES
     * computes "input" signal value
     * at "delay" prior to the array pointer "input" into the "input" array.
     */
    register int n;
    register float t;
    register float *f;
    register float arg1, arg3;
    register float denom;
    int offset;

    if( delay < 0){
        offset = -delay + 0.5;
        offset = -offset;
    }
    else
        offset = delay + 0.5;
    t = offset - delay;
    f = input - offset; /* center sum around f */

    denom = 2.0 / (2.0 * fl + 1.0);
    *output = 0.0;
    for( n = -fl; n<=fl; n++){
        arg1 = PI * factor * (t-n);
        arg3 = PI * (t-n);
        if( arg1 < 1.e-2 && arg1 > -1.e-2)/* just copy */
            *output += factor * *(f+n);
        else /* sinc function multiplied by hamming window */
            *output += factor * (0.54 + 0.46 * cos( arg3 * denom )) *
                *(f+n) * sin( arg1) / arg1;
    }
}

```

[0111]

【表8】

Nov 16 09:07 1992 testbound.c Page 1

```

/*
 * testi_ubound - test if argument a exceeds int boundary b and print text
 */
void testi_ubound( a, b, text)
int a;          /* input: value to be tested */
int b;          /* input: boundary value */
char *text;     /* input: program name */
{
    if( a > b ){
        printf("\n%s-f-value exceeds range %d > %d\n", text, a, b);
        exit(10);
    }
}
/*
 * testi_bound - test if argument a exceeds range b1,b2 and print text
 */
void testi_bound( a, b1, b2, text)
int a;          /* input: value to be tested */
int b1,b2;      /* input: boundary values */
char *text;     /* input: program name */
{
    if( a < b1 ){
        printf("\n%s-f-value exceeds range %d < %d\n", text, a, b1);
        exit(10);
    }
    else if ( a > b2 ){
        printf("\n%s-f-value exceeds range %d > %d\n", text, a, b2);
        exit(10);
    }
}
/*
 * testf_bound - test if argument a exceeds range b1,b2 and print text
 */
void testf_bound( a, b1, b2, text)
float a;        /* input: value to be tested */
float b1,b2;    /* input: boundary values */
char *text;     /* input: program name */
{
    if( a < b1 ){
        printf("\n%s-f-value exceeds range %f < %f\n", text, a, b1);
        exit(10);
    }
    else if ( a > b2 ){
        printf("\n%s-f-value exceeds range %f > %f\n", text, a, b2);
        exit(10);
    }
}
/*
 * testd_bound - test if argument a exceeds range b1,b2 and print text
 */
void testd_bound( a, b1, b2, text)
double a;       /* input: value to be tested */
double b1,b2;   /* input: boundary values */
char *text;     /* input: program name */
{
    if( a < b1 ){

```

[0112]

40【表9】

Nov 16 09:07 1992 testbound.c Page 2

```

        printf("\n%s-f-value exceeds range %f < %f\n", text, a, b1);
        exit(10);
    }
    else if ( a > b2 ){
        printf("\n%s-f-value exceeds range %f > %f\n", text, a, b2);
        exit(10);
    }
}

```

[0113]

50【表10】

Nov 16 09:07 1992 getcrit.c Page 1

```

#include "macro.h"
/*
 * getcrit - compute error between excitation and shifted residual
 */
void getcrit( criterion, residual, exctation, shift, length)
float *criterion;      /* output: error criterion */
float *residual;        /* input : residual signal */
float *exctation;       /* input : reference signal */
float shift;           /* input : shift */
int length;            /* input : vector length */
{
    void bl_intrp();
    float output;
    register int i;

    *criterion = 0.0;
    for( i=0; i<length; i++){
        bl_intrp( &output, residual+i, shift, 0.9, 8);
        *criterion += output * exctation[i];
    }
}

```

【0114】

【発明の効果】本発明によれば、合成による分析の符号 20 化において、必要帯域幅を減らすことができる。また、本発明によれば、原信号の代りに、または、原信号使用の補充として、原信号を変化させた試原信号を使用するので、符号化誤差および必要ビットレートが削減される。音声符号化において、符号化誤差が小さくなれば、長時間予測器遅延情報の伝送頻度を低くすることができ、再構成音声品質の低下がほとんど無い、または、全く無い、遅延値補間が可能となる。本発明は、特に、無線電話網（例えば、セルラー式）および通常電話網のような音声情報通信網に適用されうる。

【図面の簡単な説明】

【図1】本発明の一実施例のブロック線図である。

【図2】開ループ遅延値を推定する相関過程に使用される標本の窓を示すタイミング図である。

【図3】図1に示された実施例に使用される遅延値の時間関係を示すタイミング図である。

【図4】適応コードブックプロセッサの一例を示すブロック線図である。

【図5】図1に示された実施例の適応コードブックの動作について標本時刻関係を示すタイミング図である。 40

【図6】図1に示された実施例の時間軸シフトプロセッサの一例を示すブロック線図である。

【図7】図6に示された時間シフトの動作について初期条件集合の一例を示すタイミング図である。

【図8】図6に示された時間シフトの動作を示すフローチャートである。

【図9】図6に示された時間シフトの動作を示す、図8に連続するフローチャートである。

【図10】時間シフトにより試原音声信号を生成するのに使用される原音声セグメントを示すグラフである。

【図11】本発明の他の実施例を示すブロック線図である。

【図12】原信号と時間シフト済信号との同時性に関し、遅延値推定器の動作を説明する有限状態機械を示す説明図である。

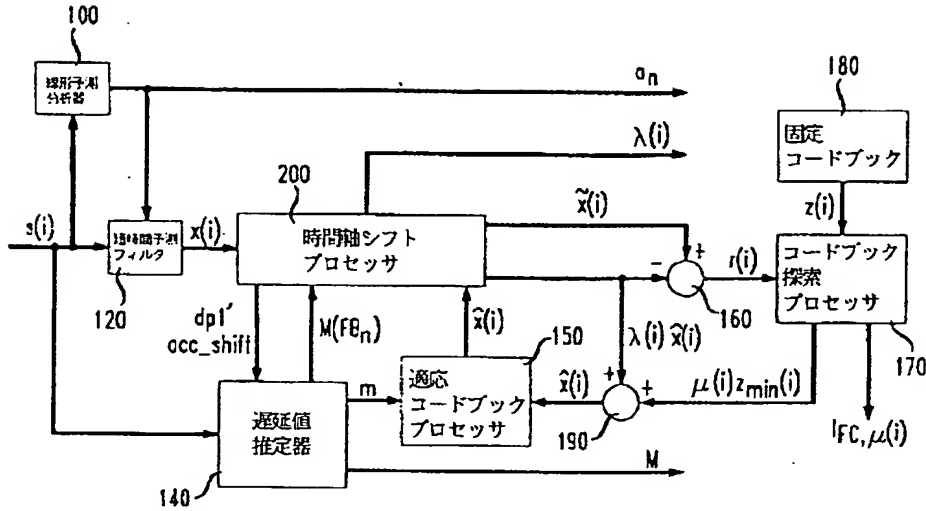
【図13】図1および図11に示された符号器の実施例に使用される受信器／復号器のブロック線図である。

30 【図14】従来の符号励振形線形予測符号器のブロック線図である。

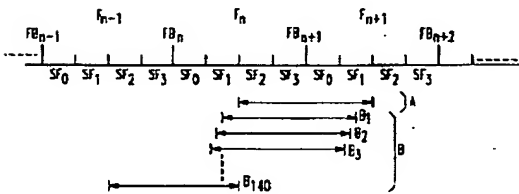
【符号の説明】

100 線形予測分析器
 120 短時間予測フィルタ
 140 遅延値推定器
 150 適応コードブックプロセッサ
 170 コードブック探索プロセッサ
 180 固定コードブック
 200 時間軸シフトプロセッサ
 40 610 試信号生成器
 620 符号（合成）器
 700 固定確率コードブック
 730 短時間予測フィルタ-1
 750 適応コードブック
 760 補間プロセッサ

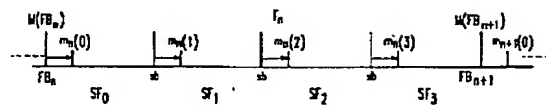
【図1】



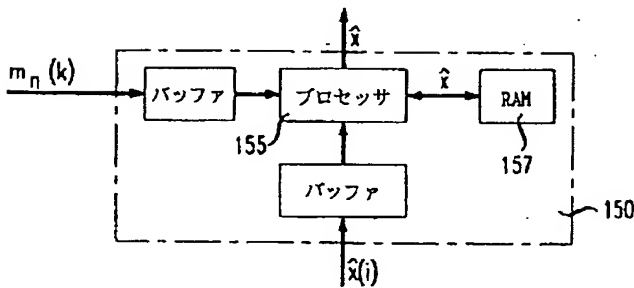
【図2】



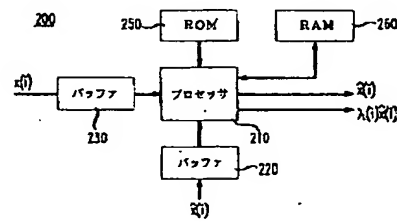
【図3】



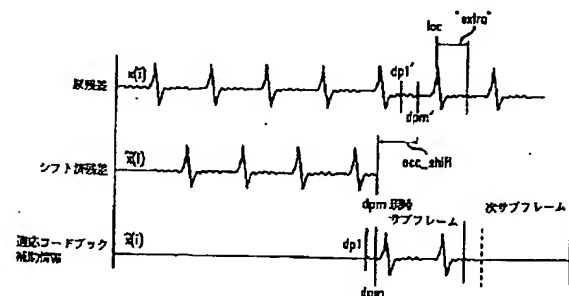
【図4】



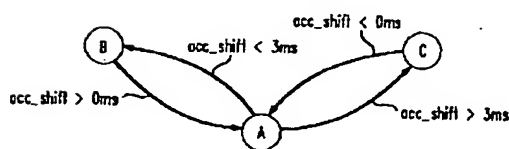
【図6】



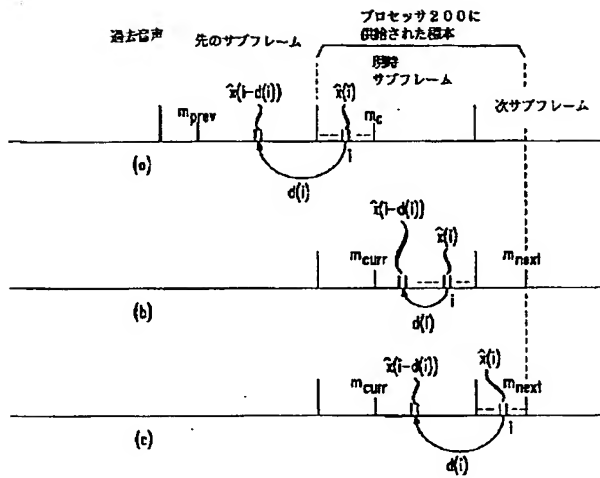
【図7】



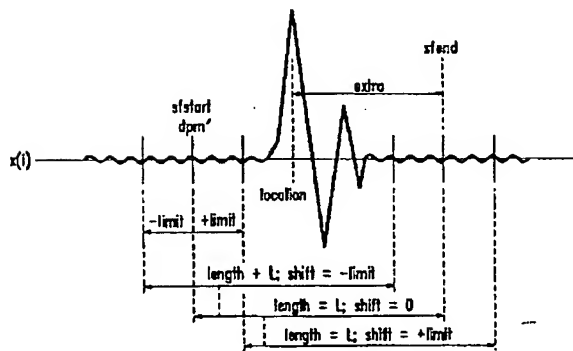
【図12】



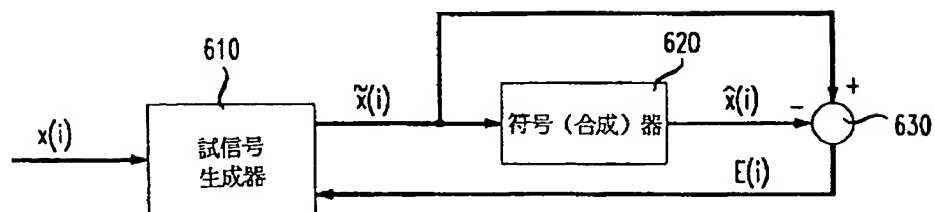
【図5】



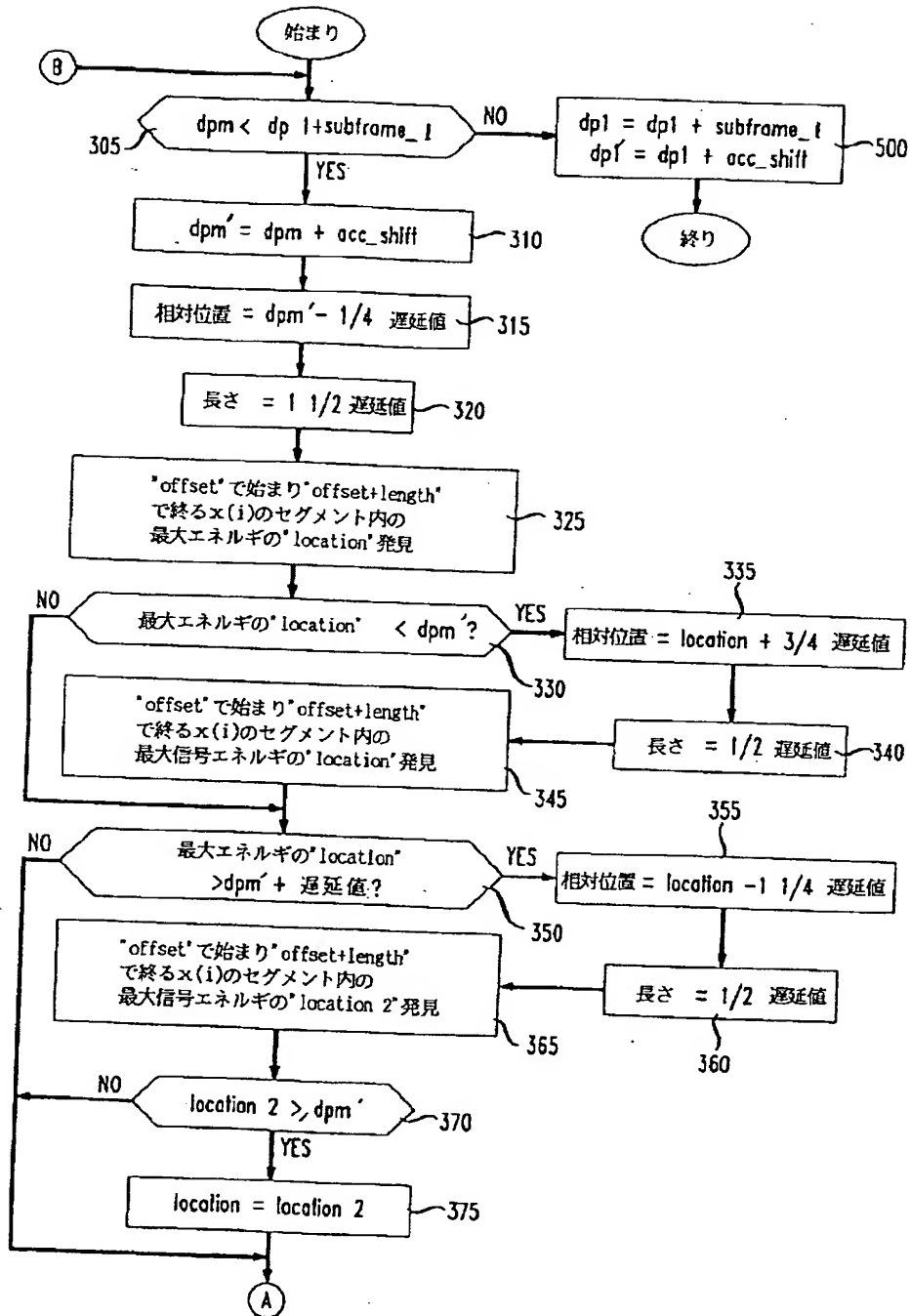
【図10】



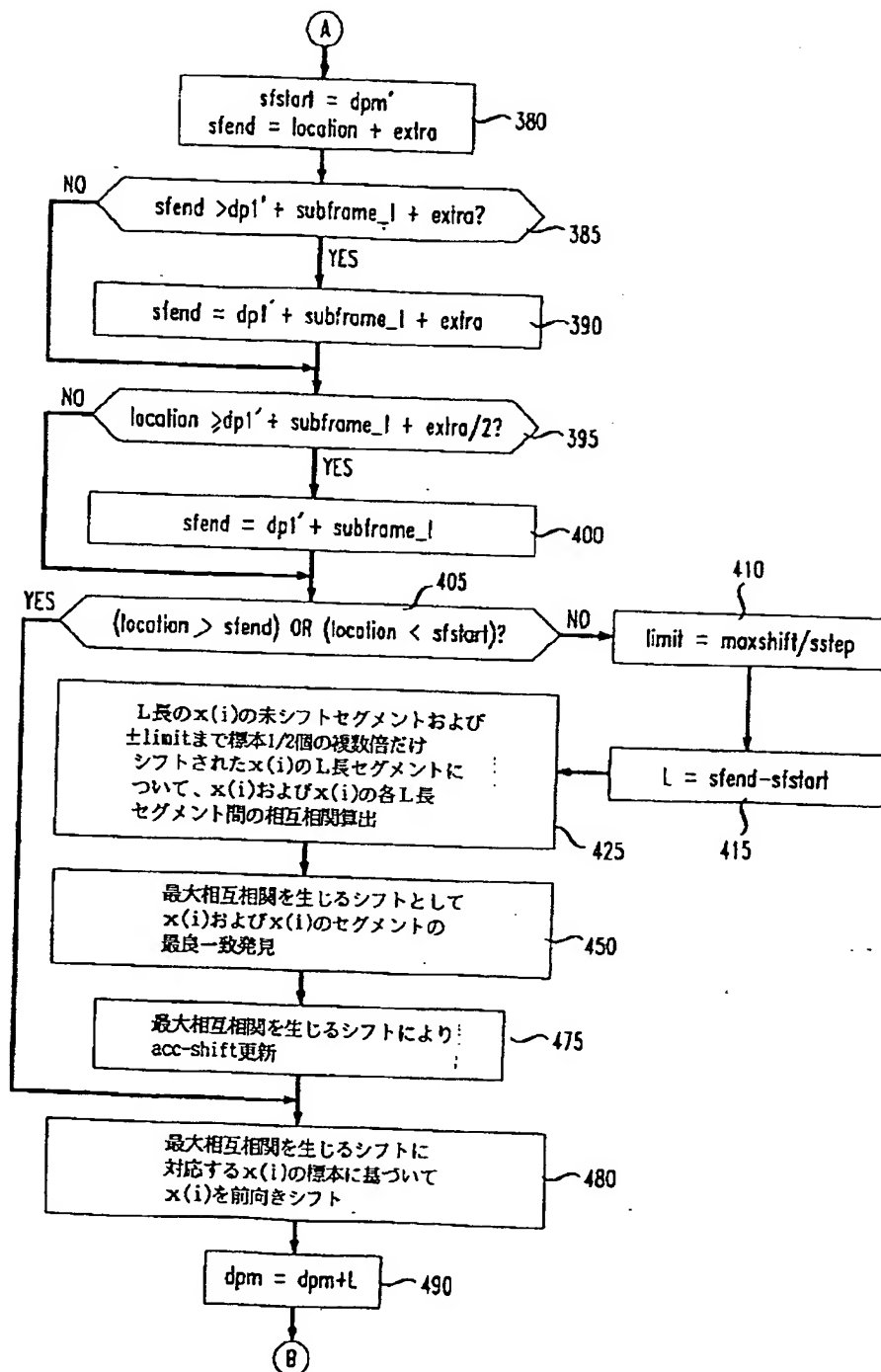
【図11】



【図8】



【図9】



[illegible]

